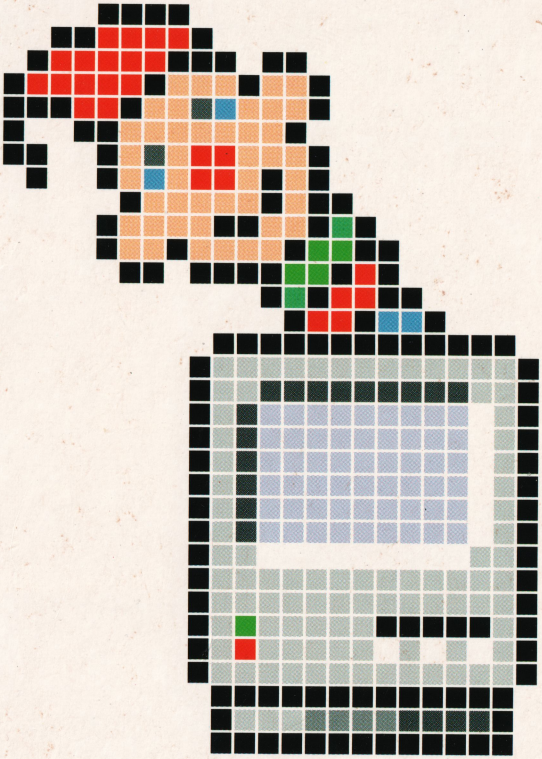
CUSTOMISE YOUR MAC WITH OUR

RESEDIT PRIMER

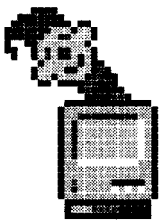


BY HERB GOODMAN

MACFORMAT

A ResEdit primer

BY HERB GOODMAN



A MACFORMAT book

HMG™ ResEdit Primer is Copyright © 1990, 1991, 1994 Herb M Goodman. All trademarks, registered names and other devices which are possessions of legal entities other than Herb Goodman are the property of their various owners, and no intention, explicit or implicit, of theft is made by their usage in the document.

This edition published by Future Publishing and issued free with MACFORMAT April 1994. Mechanical Copyright © 1994 Future Publishing.

MACFORMAT, the truly practical Macintosh magazine, is published on the first Tuesday of every month. For details of subscriptions, see page 64.

■ Disclaimer

While every effort has been made to test the instructions in this manual, under a wide variety of conditions, keep in mind that it's always a good idea to maintain current backup copies of your valuable data before trying out any new software program, whether you've paid for it or not.

Future Publishing and Herb Goodman make no warranty, either express or implied, regarding this book, its effectiveness or usefulness. They release themselves from liability for any damages resulting from the use or abuse of this manual or of *ResEdit* and such liability rests entirely on the user.

Apple Computer Inc ('Apple') makes no warranties, express or implied, including without limitation the implied warranties of merchantability and fitness for a particular purpose, regarding *ResEdit*. Apple does not warrant, guarantee or make any representations regarding the use or the results of the use of *ResEdit* in terms of its correctness, accuracy, reliability, currentness or otherwise. The exclusion of implied warranties is not permitted by some states. The above exclusion may not apply to you.

In no event will Apple, its directors, officers, employees or agents be liable to you for any consequential, incidental or indirect damages (including damages for loss of business profits, business interruption, loss of business information, and the like) arising out of the use or inability to use *ResEdit*, even if Apple has been advised of the possibility of such damages. Because some states do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitations may not apply to you. Apple's liability to you for actual damages from any cause whatsoever, and regardless of the form of the action (whether in contract, tort (including negligence), product liability or otherwise), will be limited to $50.

Contents

Introduction	Why I wrote this manual Electronic mail	**5**
Before you begin	Warning on using *ResEdit*	**6**
Resources	Adding keyboard equivalents Brief tour of various resources	**7** **10**
Adding program icons	Step by step procedure for adding and creating icons for programs	**14**
Animated cursors	Create your own animated cursors	**24**
Desktop alterations	Customise your desktop Desktop icons versus generic icons	**33** **37**
Get Info comments	How to save and restore your comments to the Get Info boxes	**40**
File Info window	Finder flags in the File Info box	**47**
Menu editing	Detailed discussion with examples	**49**
Bits & pieces	Sound (snd) resources Custom mouse tracking Version (vers) resources	**61**

WARNING!

ResEdit is a very powerful program which can easily damage both programs and System software. Only ever use it on copies of programs, and be prepared by having a backup and emergency startup disk.

You use the software and suggestions in this book entirely at your own risk. **MACFORMAT** and Herb Goodman cannot be held responsible for any consequences of using *ResEdit*. You have been warned!

Introduction

As everyone knows, the Macintosh is a very versatile and productive computer. But all work and no play makes life dull. Apple gave us the interface with cursors, menus, icons, dialogues, etc, but after a while they can get boring. However, Apple also gave us the 'tool' to change these items (AKA resources) to our own taste and liking with a program called *ResEdit*. The problem is, there is very little documentation available on how to use this program. The documentation that does exist is mostly obscure and has to be sifted out of (pieced together from) hundreds of pages of the *Inside Macintosh* manuals (which aren't cheap).

Though I'm not a programmer, I feel fairly knowledgeable and comfortable using *ResEdit* to modify existing applications and files (including system files like the Finder, System and Desktop). It has taken me quite a long time to get to this stage, since I had no one to teach me. Most of my knowledge has come from trial and error (with many system crashes). That's why I decided to put together an easy, but comprehensive, manual on *ResEdit*, to help you unravel the mysteries of the Mac.

In some cases there will be duplication of information from one chapter to another, depending upon the topic under discussion. If the novice user wants some specific information on how to perform a given task, they should be able to find it in a particular chapter, rather than having to piece it together from various chapters.

This is the sixth release of this manual.

NOTE: This manual is based on Version 6.x of the Macintosh System Software, but most of the principles discussed apply equally to System 7.x. We recommend that if you intend to modify System 7.x resources, you familiarise yourself with the detailed documentation in the latest edition of the Apple Technical Library 'Inside Macintosh' series published by Addison-Wesley.

■ Electronic Mail:
I can be contacted on the following BBSs:

CompuServe:	73300,3273
GEnie:	H.M.GOODMAN
America Online:	HMG

Before you begin...

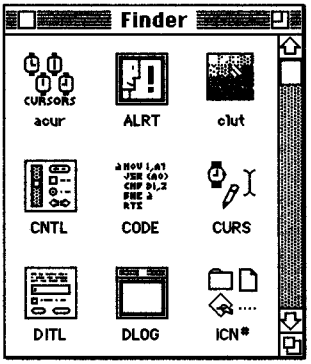
ResEdit is a very powerful program. Because it's so powerful, it can be very dangerous, often lethal, if improperly used. Therefore, it is imperative that you understand and adhere to the following rules and guidelines before and after using *ResEdit:*

1. Make a copy of the application or item that you intend to modify and place it in another folder, volume or disk.
2. **ALWAYS WORK ON A COPY, NOT THE ORIGINAL!!!**
3. **NEVER DELETE, DISCARD OR TRASH THE ORIGINAL UNTIL YOU ARE SURE THAT THE COPY WORKS, THE WAY IT WAS INTENDED TO WORK!**
4. Read rules 1, 2 and 3 again!

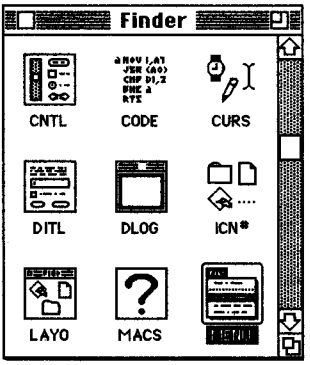
If you are planning to modify a system file, such as the System or Finder, remember rules 1–3. Then, move the original(s) to the Desktop or to another folder. After modifying the copy(ies), rename the copies to the original name(s) and move the copy(ies) into your System Folder and reboot (restart). If everything goes well, you can delete or trash the originals (but *always keep an unmodified backup* in case problems become evident after some time). If you have a problem, boot up from another disk and move the modified copies from the System Folder and put the originals back in. You may then reboot off of your normal startup volume and trash the copies you modified.

Adding keyboard equivalents to MENUs

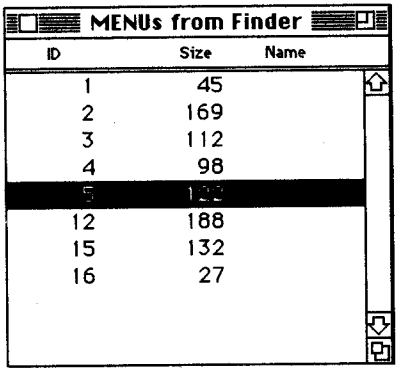
The mouse is one of the most creative devices for input today. Unfortunately, sometimes it's easier to simply use the keyboard. Here's where *ResEdit* comes in... With *ResEdit,* you can add keyboard equivalents to menu items. Simply choose an application to edit. Launch *ResEdit,* and open the application via Open from the File Menu (you will be presented with the standard Open Dialogue Box). Navigate until you find the application you want to modify and double click on it to open it. You are now presented with a window containing all of the resources contained in the application. They are identified by icons.



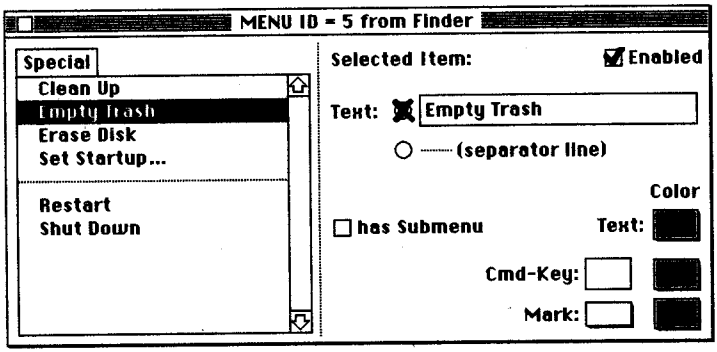
Scroll down the resource list until you find the MENU icon.



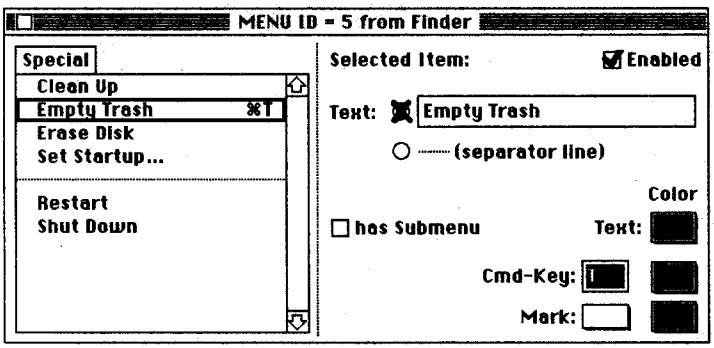
Double click the application's MENU icon. You'll get a list of the menu ID numbers in the application.



Find the menu you want by double clicking on the ID numbers one at a time until the 'title' string is the name of the menu you want. (Occasionally, a program will have a few copies of the same menu. You should put your key equivalents in each menu with the right name.)



Select the command you want to add the keyboard equivalent to. Decide what the control key will be. (Make sure that there are no others with the same one or something unexpected may occur. *ResEdit* gives no warnings about this.)



In the Cmd-Key box, type in your command-key equivalent. Now, simply close the windows and save everything. The next time you load the program, your key equivalent will automatically appear next to the menu selection. You simply have to depress [Command] [your key] to activate it!

ResEdit types

As you know, each resource has its own type (by which we mean variety). This is a four character string which distinguishes that type of resource from any other. Many resource types are considered to be a 'standard' (as defined by Apple). Other resource types are individually created by programmers and developers. I'll list for you, and explain, all the important resources for which *ResEdit* contains templates. (A template is a resource of type TMPL, used by *ResEdit* to know how to interpret each resource type.)

ALRT (ALeRT)

This resource type is an alert box. It is basically a dialogue box that can have only two buttons, preferably OK and Cancel. An alert box will report to the user any error the program may have created. Opening a resource of the type ALRT will give you a condensed screen image of how the alert will look. Double-clicking on the alert will zoom in and allow you to edit its text.

BNDL (BuNDLe)

The information contained here basically holds together the data that the Finder will need to display the icons within the application. (It may also contain a pointer to the version data of the application.) Any application that has an icon will have a BNDL containing at least one pointer to a ICN# (Icon List) and one to a FREF (File Reference). The ICN# contains the picture, while the FREF contains the file types the icons will go to.

CNTL (CoNTroL)

A control is any non-graphic, mouse-modifiable item in a dialogue box or window. OK buttons, check boxes, radial buttons (little circles), and scroll bars are all common examples of controls. A control will have its location in the window, its initial value, whether or not it's visible, its maximum and minimum values, its type (in the form of a number), a possible pointer to more data, and its title (the text that appears in, on, or next to it).

DITL (Dialog ITem List)

A resource of type DITL will contain the layout of a dialogue box, including the positioning of the controls, text and pictures. The contents of the dialogue box will show up with dimmed rectangles around each item. Double-clicking on any of those will bring up a box describing the particular control, including its type, position, vis-

ibility and text content.

DLOG (DiaLOG box)

This is the dialogue box itself, with its DITL displayed. Opening one of these will show a miniaturised version of this dialogue on the screen. As with an alert, double-clicking on the box will allow you to edit its DITL.

DRVR (DRiVeR)

These are, most commonly, desk accessories. A driver resource contains the hexadecimal representation of the DA itself; it should not be edited unless you understand machine language.

FOND (FONt Data) and FONT (FONT)

A FOND resource is a resource that contains pointers to the FONT resource. It tells the program what fonts it has available. A FONT resource defines the actual characters which will appear on the screen. Double clicking on a FONT resource will show ASCII character 65, which is the capital letter 'A' in most fonts.

FREF (File REFerence)

This type of resource contains the pointers to the correct ICN# which will show for the displayed TYPE. Every application with an icon will have at least one FREF with the type of APPL, and an ID number pointing to an ICN# which will be displayed for the application.

MBAR (Menu BAR)

This resource will hold an entire menu bar. It is actually a list of MENU resources, in order from left to right across the screen. Most applications do not use MBAR resources, but instead simply put individual menus up one at a time. If there is one present, switching two entries would switch those menus' positions in the menu bar.

MENU (MENU)

This resource holds an individual menu. In old versions of *ResEdit*, each menu item is completely editable, but it took forever to scroll and open them. In the newer versions, the basic menu items are editable, and it is considerably faster. Most of the important items which are editable here were discussed above.

PICT (PICTure)

PICTs are the standard for graphics on the Macintosh, just as TEXT is the standard for written items. A PICT resource is not editable

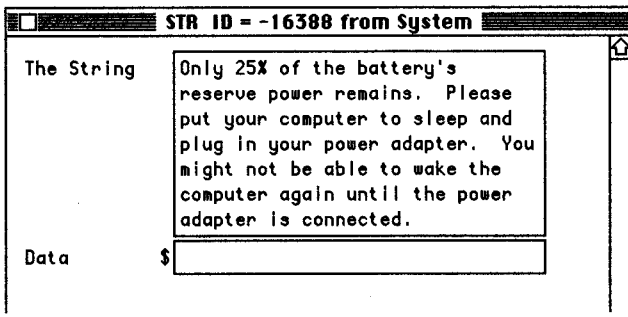
through *ResEdit*. The actual picture may be bigger than can be displayed in a standard *ResEdit* window. In order to get the full picture (sorry about the pun), you can COPY the picture while it's displayed and PASTE it into your scrapbook.

SIZE (SIZE)

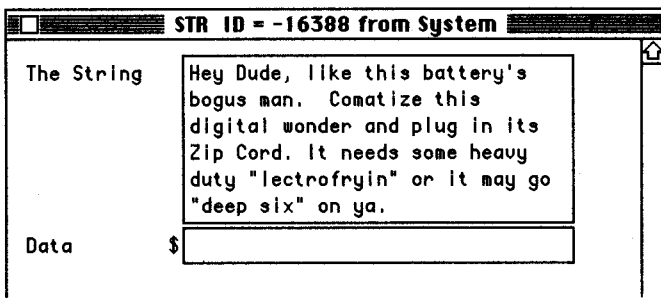
This resource, originally intended to hold the size of an application under Multifinder, evolved to include all of an application's attributes under MultiFinder. Most applications that have been run under MultiFinder will contain a SIZE resource of -1.

STR (STRing)

This is rather obvious. It's a string that the application will probably need at some point. Changing these is no problem and can be quite fun.


STR ID = -16388 from System
The String
Only 25% of the battery's reserve power remains. Please put your computer to sleep and plug in your power adapter. You might not be able to wake the computer again until the power adapter is connected.
Data
$

Before


STR ID = -16388 from System
The String
Hey Dude, like this battery's bogus man. Comatize this digital wonder and plug in its Zip Cord. It needs some heavy duty "lectrofryin" or it may go "deep six" on ya.
Data
$

After

STR# (STRing list)

A STR# is a resource which contain a whole bunch of strings. Each of these can be edited as if they were individual STR resources.

vers (versION)

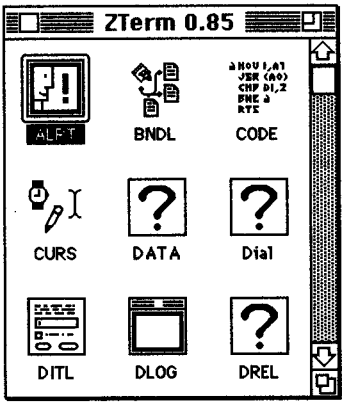
This resource will contain the version information about the application. This will contain the creation, modification date, and what will appear in the Get Info box. (Note: If there is a BNDL pointing to a resource of ID 0 then that will be displayed in the Get Info box if the vers resource is missing.)

WIND (WINDow)

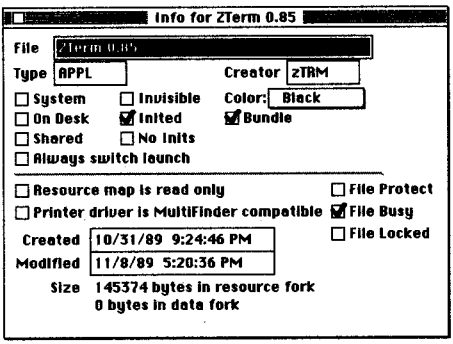
This is a resource which defines the position, size and type of window. A miniaturised version of the window will appear when this type is selected. You can move the window to another position with the mouse if you want.

Adding icons to programs

Launch *ResEdit* and open the application via Open from the File Menu (you will be presented with the standard Open dialogue box). Navigate until you find the application you want to modify and double click on it to launch it. You will now be presented with a window graphically listing all of the resources contained in the application. They are identified by icons (if 'by Type' is selected from the View menu).

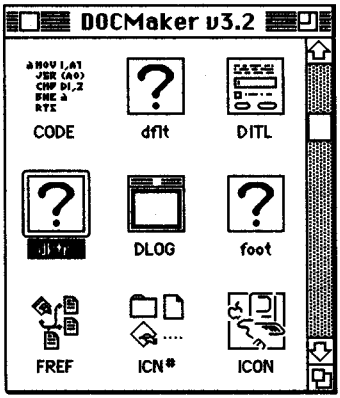


Choose Get Info for <filename> from the File menu and look at the information box which is presented. At the right side, under the file name, there is a box labelled Creator. If there is anything besides a blank or a '????', write down the creator's name EXACTLY as it appears. Close the info box.



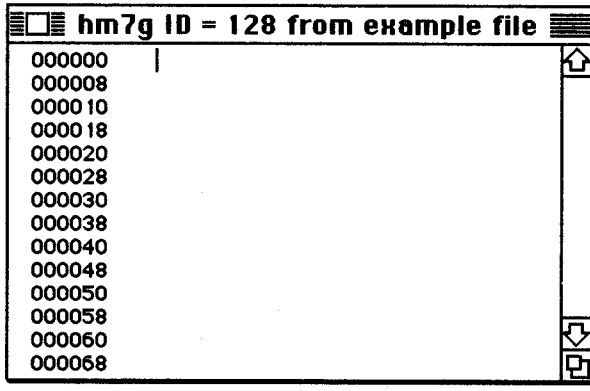
MAKING THE PRIVATE RESOURCE:

In order for the Finder to recognise an application as other than generic, it must have a 'private resource' (aka signature resource). This is simply a resource with the same name as that of the creator.

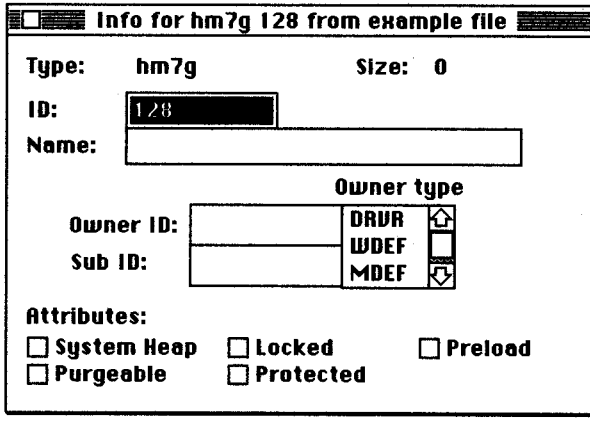


If you found a creator's name earlier, look for it in the list of resources. If there is not one, you must choose one. It can be any four character name, but be careful that it does not conflict with the creator's name in any existing (or future) applications. Sound tough? Try some obscure combination using mixed caps (caps are not mandatory) and lower case, maybe with a number thrown in, and the chances are that it will be unique (like H7mg or hm7g). If there is an accidental duplication, the icon for this will be used for the other application also (or vice-versa, depending upon precedence).

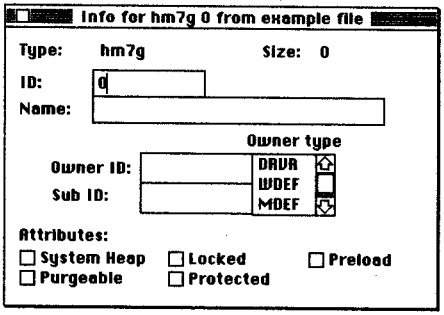
Having thought of a name, choose Create New Resource from the Resource menu. This time, using the keyboard, enter the 'creator's' name and *ResEdit* will make a resource of that name, assign it an ID#, and present you with an open data box. Notice that it doesn't have to contain anything.



Close the data box, select the ID# resource and choose Get Resource Info from the Resource menu.

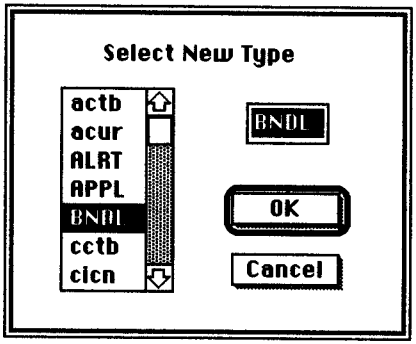


Edit the Resource ID box to make the resource ID = 0. Close the resource.

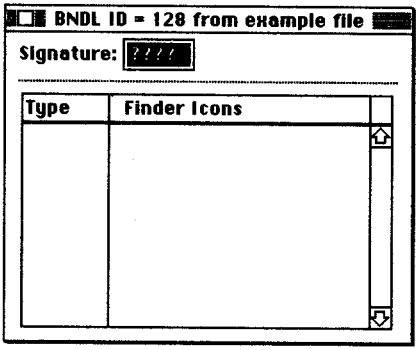


MAKING A BUNDLE:

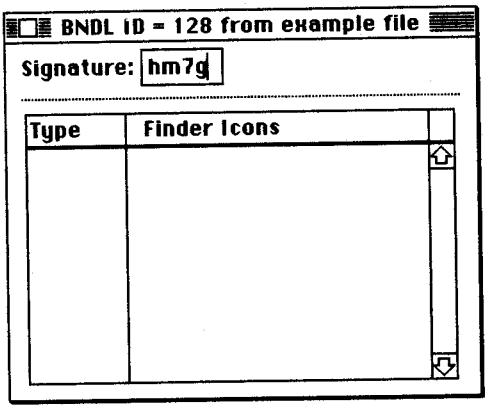
From the Resource menu, select Create New Resource.



Select the type BNDL. This will automatically create a new BNDL resource and open the resource window.

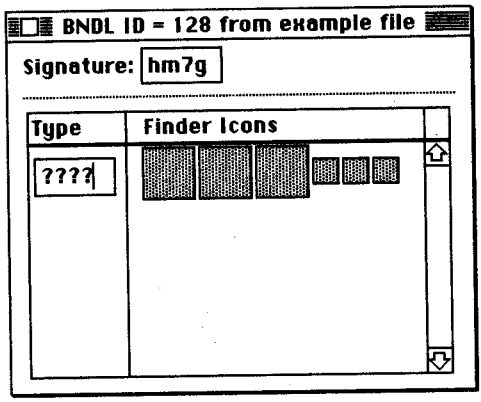


Now, in the signature box, type in the signature (name) you created previously.

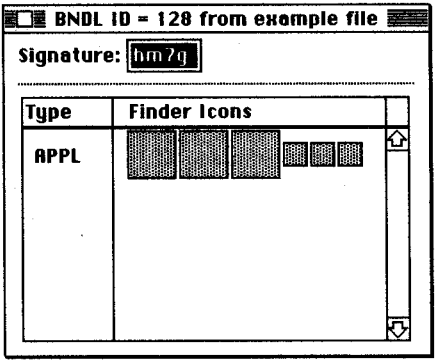


MAKING THE FREF RESOURCE:

From the Resource menu, select Create New File Type.

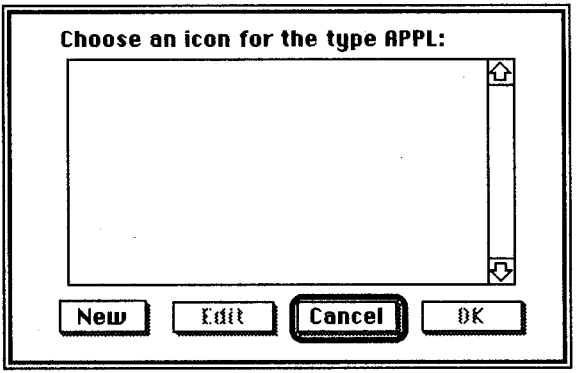


ResEdit has actually created a FREF resource for you. All you have to do is assign a type to it. Select the box with the ???? and enter in 'APPL' which stands for 'application'.

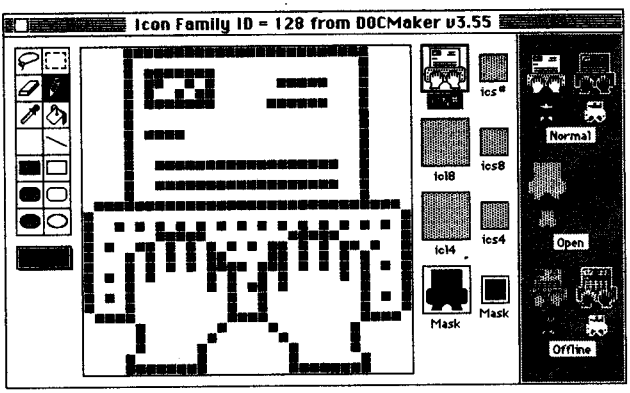


MAKING THE 'ICN#' RESOURCE:

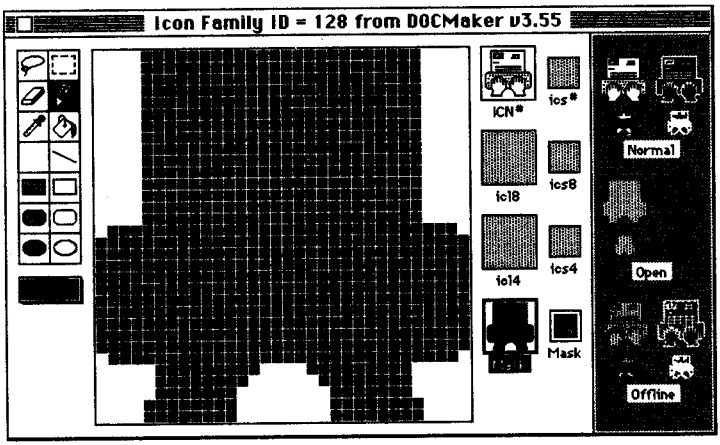
From the BNDL window, double click on the leftmost box under Finder Icons. You will be presented with the following dialogue box:



Select New, and *ResEdit* will create a new ICN# and open the box in which to create (draw) your new icon. The ID number of the created resource will be 128 (the default).



ICN# Button selected

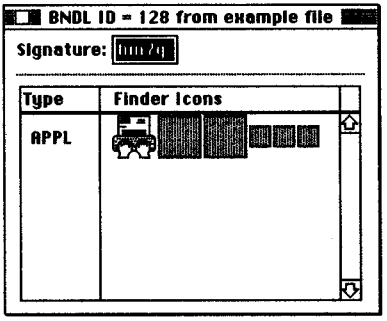


Mask button selected

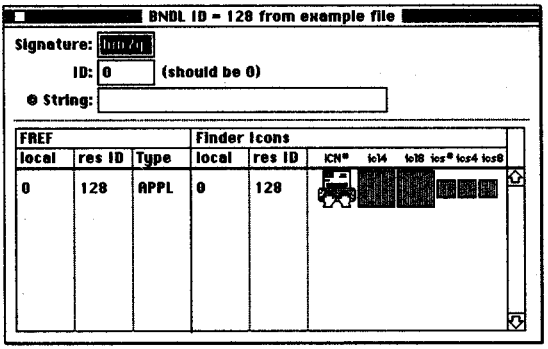
The large panel is your edit window (work area). To the left of the edit window is your tools palette. To the right of the edit window are selectable buttons that determine the type of ICN# you are going to create or edit. The frame to the extreme right is used as a visual guide to display your icon as it will appear on the desktop or other

backgrounds (you can change backgrounds via the Icons menu), in unselected and selected state, in normal and reduced size (sicn), in normal mode, open mode (application is launched/file is open), and offline mode (volume/disk containing your icon is ejected but still mounted on the desktop).

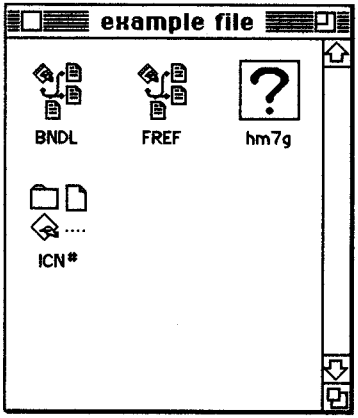
Start by clicking the ICN# button. Create your icon in the edit window using the various tools and menu selections available. Once you are satisfied with your work of art, choose the marquee tool and grab the entire frame (or choose 'Select All' from the Edit menu). Now choose Copy from the Edit menu. Click on the Mask button and choose Paste from the Edit menu. Fill in any desired areas with the paint can. Once you have finished your creation, close the ICN# window. Now you can see the result...



BNDL – normal view

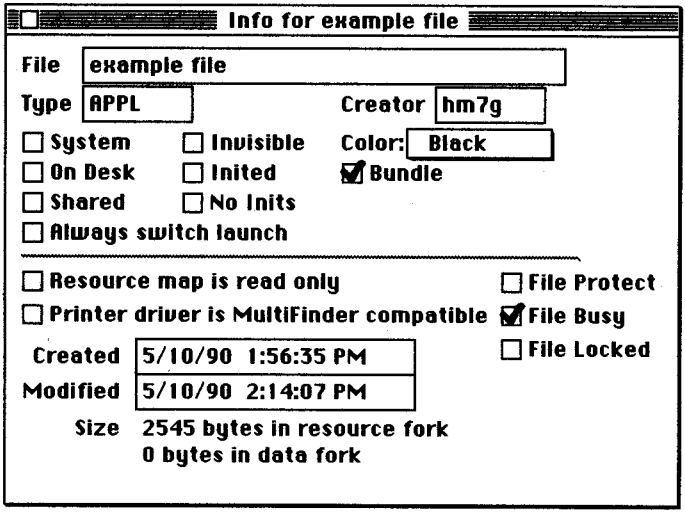


BNDL – extended view (from BNDL menu)



FINISHING UP

Close all of the resources. From the File menu, select Get Info for <filename>.



Enter your signature resource name in the space creator. Click on the selection 'Bundle' until the 'X' appears, showing that the bundle bit is set. Click on the 'Inited' box until the 'X' disappears. At this

point, only the 'Bundle' box should have an 'X'. Close the box; you will get a chance to save these changes. Save all of your changes and Quit *ResEdit*.

If this is an icon added to a file which previously didn't have one, the new icon should appear when you exit from *ResEdit*. If you are changing an existing icon, you will find that nothing has happened. This is because the Finder thinks that it already has the correct icon. The Finder keeps a copy of all icons which it runs across and stores each of them singly in the Desktop File, and it will not look for a new one if the creator is the same. There is an easy remedy for this situation:

First: take a newly formatted (or erased) floppy disk and copy your application to it. Then drag the floppy disk to the Trash (to dismount and eject it).

Second: while holding down the [command] and [option] keys, insert your floppy disk into the drive. You will get an ALERT dialogue box asking you if the Desktop should be rebuilt on that disk. Click Yes. After the operation is finished, the floppy will be mounted on your Desktop. Your new icon should appear in it.

Third: copy the application from the floppy disk to your original disk and answer Yes in the Replacing Items dialogue box. The old icon (generic) should be replaced with the new icon you designed.

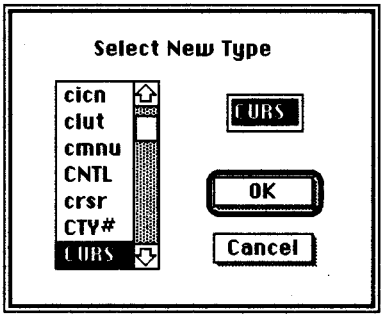
Animated cursors...

The following information is discussed in this chapter:
- How to make animated cursors
- Resources covered: acur, CURS.

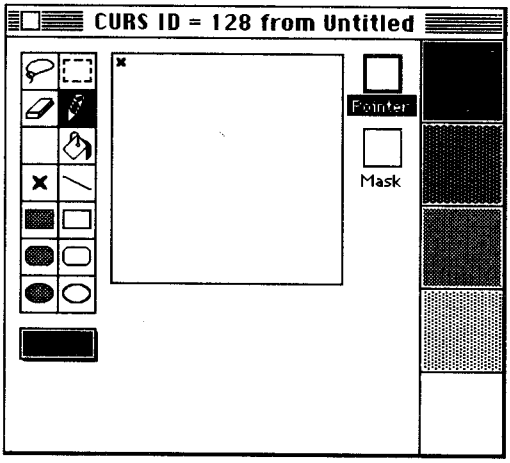
• CURS (CURSor)
The CURS resource contains the little graphic display you get when the Mac is doing something like loading a program, copying a file, saving information.. You've probably seen many examples of this such as a watch with rotating hands, a spinning beach ball, an arrow (for pointing and selecting), an I-beam for insertion point as well as others. You can find CURS resources in almost every application.

• acur (aNIMATED curSOR)
The acur resource is what makes a series of cursors sequence through. It tells the Finder how many cursors there are in the sequence, their execution order, and how fast to frame through them. I have only found the acur resource in the Finder. Most applications do not use this resource, but have the information about their cursors buried in their code. That doesn't mean that the cursors can't be changed. They can, as long as the information about the cursors remains the same. You'll understand later in this chapter.

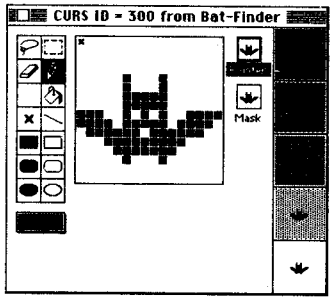
LET'S MAKE AN ANIMATED CURSOR...
To start with, launch *ResEdit* and from the File menu (this depends on how you have your preferences selected in *ResEdit*), select New. You will be presented with a standard Open dialog box. Name the file and click OK. You now have a new open window with nothing in it. From the Resource menu, select Create New Resource.



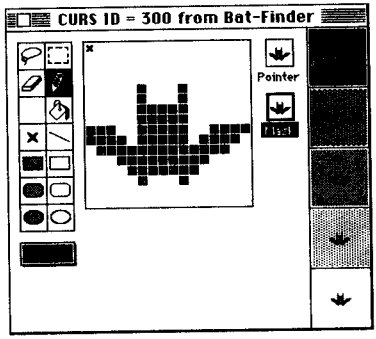
Find the CURS resource and click OK. Now you're presented with the window in which to create your cursor.



If you're used to using a Fat Bits editor, you should have no problem. Your tools palette is on the left. The centre frame is your work area. Adjacent to the work area are two buttons, one to create your cursor pointer and the other to create the cursor mask. On the extreme right are various backgrounds that will display how your cursor will appear on the desktop. The small x that appears in the work area is the hotspot point which aligns the cursor with the mouse location. You can change this location by selecting the X tool from the tools palette. From this point on, I'll use my own custom cursors and resources as examples.

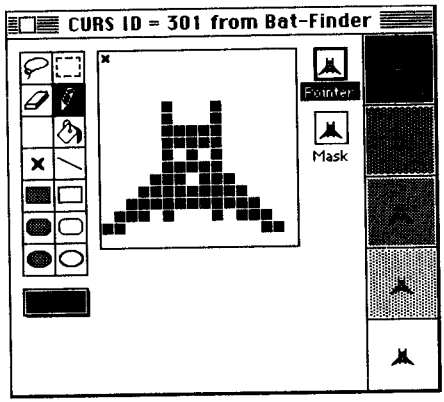


Pointer

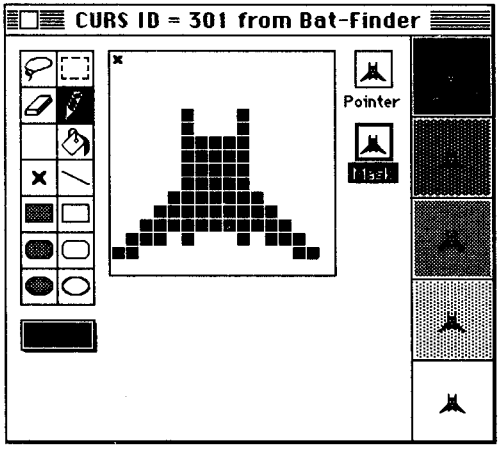


Mask

Now create your cursor. When you're satisfied choose Select All followed by Copy from the Edit menu. Click on the mask button and choose Paste PICT from the Edit menu. Fill in appropriate holes to get your desired effect. Go to the CURS menu and select Try Pointer. Your arrow will change to your cursor. Move it about the screen. Go back to the CURS menu and select Try Pointer. Now make any changes you want. When you're satisfied, choose Select All followed by Copy from the Edit menu. Then close the window. From the Resource menu, select Create New Resource and you'll get another blank cursor-editor window. Now choose Paste from the Edit menu. You should have the same cursor as the first one you created.

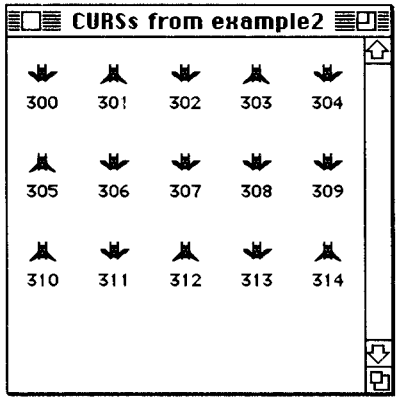


Pointer



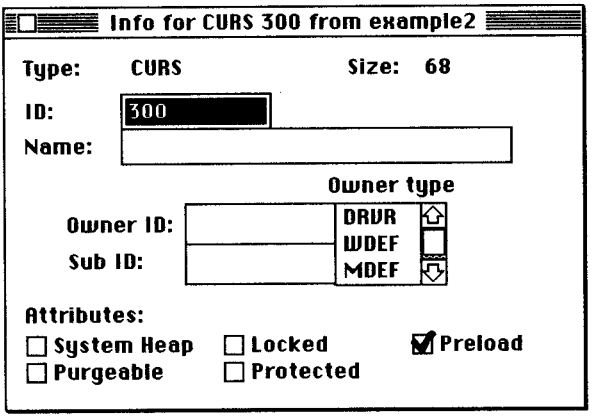
Mask

Modify this one to show a different position as I have done. Again follow the same instructions as you did before to create the first cursor. To make a sequence, double click the first, copy it, paste it in the third. Double click the second, copy it and paste it in the fourth. Do this as often as you desire, until you have completed a series as I have.



Now we need to change the information about each cursor. Select the first cursor and from the Resource menu choose Get Resource

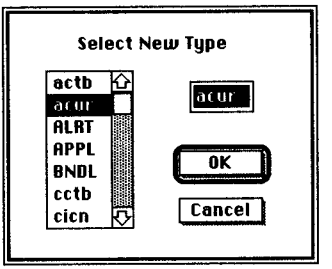
Info. You'll now have a window like this:



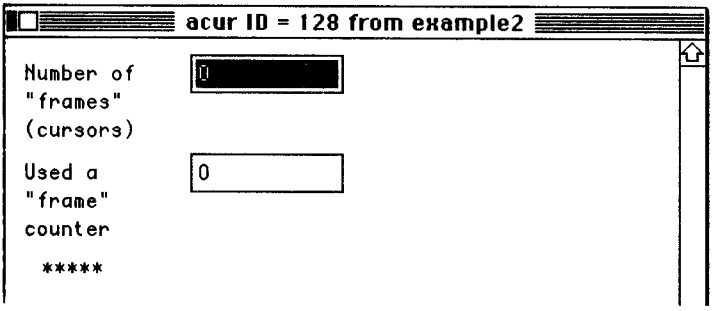
Change the ID# to anything equal to or above 256. Check the Preload box. Close the window and select the next cursor in your series. Do the same thing you just did but increment the ID# by 1 (257). Repeat this procedure for all of your cursors, each time incrementing the last number used by 1 (258, 259, 260, etc).

You've done a lot of work by now, so let's not lose it. Close all windows, saving changes when prompted. Relax for a few minutes. Grab a cup of coffee, pet the dog (or cat). When you're ready we'll start the next phase...

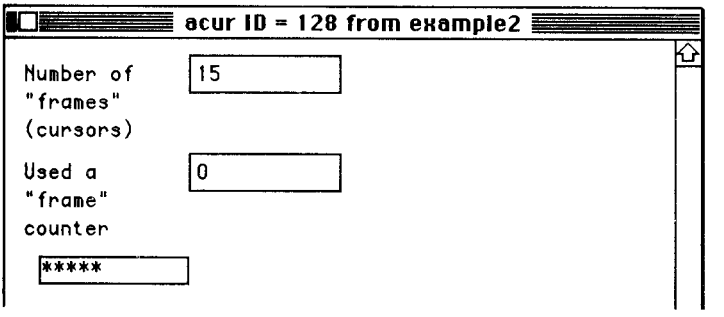
Now that you're rested, we'll begin the next phase... creating the acur resource for your animated cursor. Begin by opening your file in *ResEdit*. A window will appear showing the CURS resource. Go to the Resource menu and select Create New Resource.



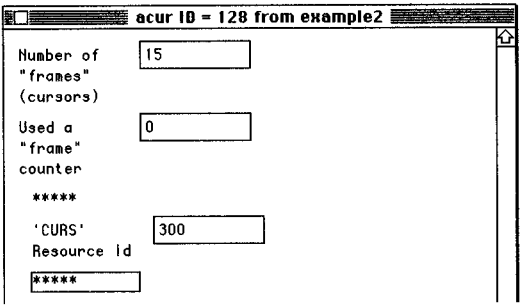
Select acur and click OK. You're now presented with the open acur resource in which to tell the Finder what you want displayed.



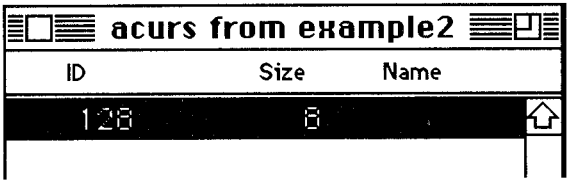
Enter in the number of cursors you have in your sequence.



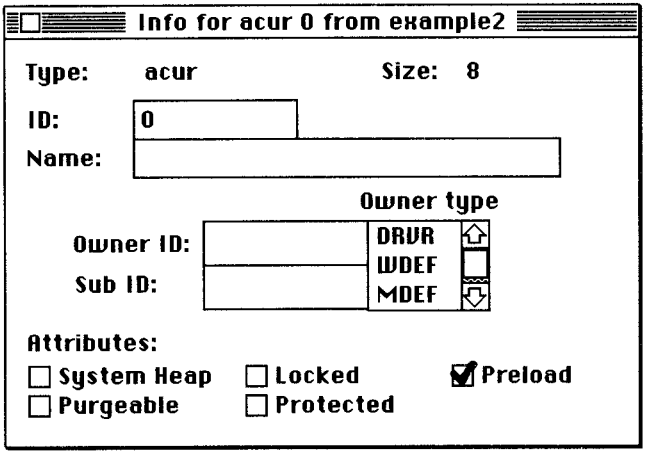
The Used a 'frame' counter tells the Finder how long to hold before switching frames (each cursor). Enter a '0' for our example (you can experiment with this later, on your own time, not mine). You see the row of *****? Select them, and from the Resource Menu, select Insert New Field(s). Another box will open below the *****. Enter the first ID# of your cursor set (256). (Note: Since I'm using my own cursor set for this example, my first curs ID# begins at 300).



Now select the next set of ***** and repeat the process. Keep doing this until you have listed your complete set of cursor ID#s. When you have finished, close the window...



Next, we need to change the information about your new acur resource. Select the acur ID = (some number or default 128). From the Resource menu, select Get Resource Info. Change the ID# to 0. Check Preload.



Finally, close all the windows and save your changes when prompted to. Quit the *ResEdit* program. You have now finished creating your resources. The next stage is to install them, so take another break...

We're ready to install your new resources. Make a copy of your Finder. I would put this copy on a blank floppy so you can't open the wrong file. Now launch *ResEdit* and open the copy of the Finder. You should see all of the resources contained in the copy of the Finder. Now open the file you created containing your new resources. While holding down the [command] key, select both your new acur and CURS resources. From the Edit menu, select Copy. Now select the copy of the Finder. From the Edit menu, select Paste. Answer Yes to replacing the same resources. After you're finished, save your changes and quit *ResEdit*.

Now open your System Folder and move your original Finder to another folder (create one naming it temp), making sure that it resides outside of the System Folder. Next move your new Finder into your System Folder and rename it 'Finder'. And finally, reboot your Mac to have the new Finder load. Your new animated cursor should now work.

Test your animated cursor by duplicating a large file or doing a disk to disk copy. You may notice from time to time that the stationary watch cursor appears. That's because that cursor is being called from the System file. You can change that too, by opening the CURS resource in the (copy of) System File, select the watch cursor and double click it. You again have the Fat Bits editor in which to change the watch to something else. Then follow the same procedures as before, with the Finder.

CHANGING CURSORS IN EXISTING APPLICATIONS

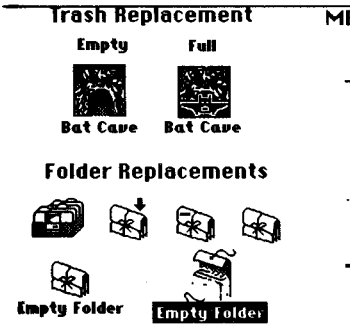
As I mentioned before, most applications which contain animated cursors do not have the acur resource. Since that information is contained in their code you can change their cursors, but you must keep the same information recorded in their cursors and apply it to your cursors.

For example, if there are only four cursors with ID#s of 300, 301, 302 and 303 with the attributes of preload and purgeable, then your cursors must contain the identical information and the same number of cursors. The safest way to change them is to use the Copy/Paste

method you used to duplicate your cursors. In other words, open your specific cursor (remember the editor window?) and Copy it. Then open the application's specific cursor and Paste in your cursor picture. Do the same for all of the cursors in the application. That's all there is to it!

Remember always to WORK ON A *COPY* OF THE APPLICATION, 'cause if you screw-up you're in deep... trouble. Or another way to put it, "the faeces has hit the rotary oscillator". Nuff said – enjoy!

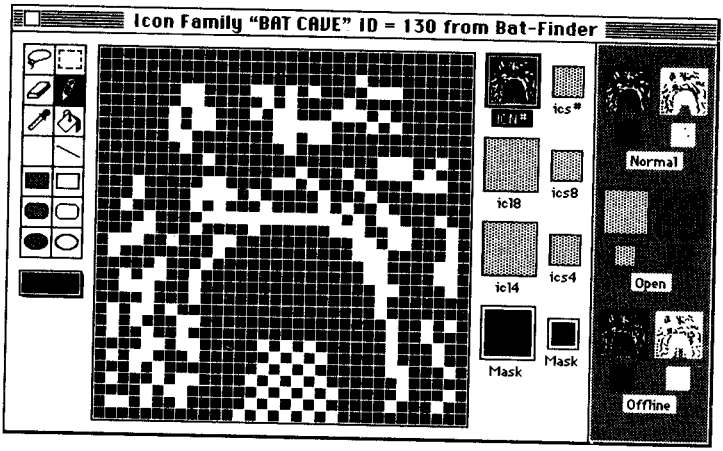
Desktop alterations...



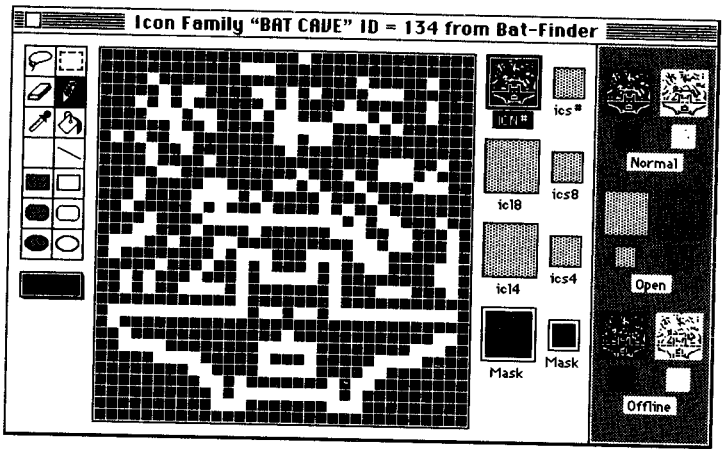
The examples above illustrate how you can change the appearance of your desktop. There are so many things you can change in the desktop and most are very easy to do with *ResEdit*. I leave the creativity to you. All of the resources you see above are ICN#s and menu items found in the Finder using *ResEdit*. If you're not an artist, you can use many of the public domain resources found on many BBSs. It's just a matter of Copy and Paste from the Edit menu of *ResEdit*. Just make sure the information pertaining to the existing resources is matched by your replacement resources (use Get Resource Info on each existing resource and modify your resource to match it), and then Copy your resource and Paste it into the existing resource. Save changes only to the Finder you are modifying.

To create new ICN#s follow the procedures you followed when making your animated cursors (see previous chapter); the only difference should be the resource type selected from the New resource window. Use the editor in the same way you did previously.

As you have noticed, there are two ICN#s for the trash can, so you must create or modify both:

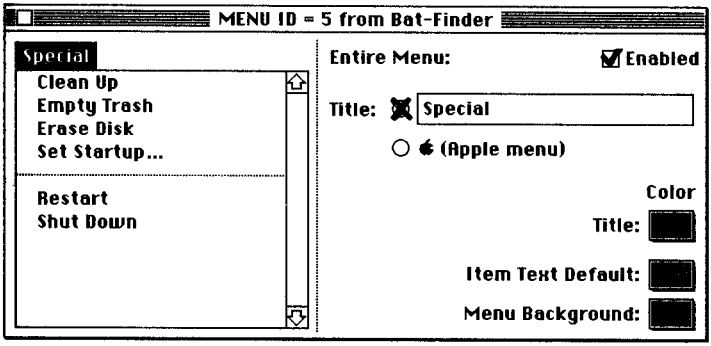


This one is for the 'empty' trash can replacement.

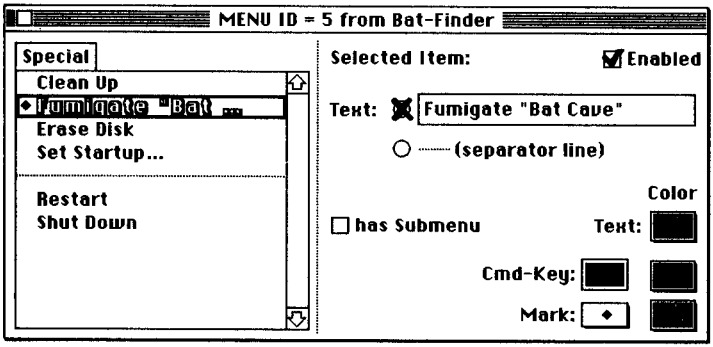


This one is the replacement for the 'full' trash can.

■ To change the menu item reflected in the Special menu (pertaining to Empty Trash) modify Menu ID#5 in the Finder:



Before changes are made...



After changes are made...

In this window, select the item (text) and type in your new text. Then select the style (Style menu) you want and colour (if you have a colour Mac). You can set a Mark (pull down menu from selecting Mark box) before the text. (A Mark character is a single character which is used to flag a menu item which has been selected by the user.

For example, most word processors which have a menu listing the available fonts, use the check-mark character to indicate which font

a user has chosen. After a menu item, with a mark character assigned to it, has been selected, the mark character will appear in the menu to the left of the menu item's text. The next time that item is selected, the mark character will be removed from the menu display. In some cases, however, the mark character will remain until another item in that menu window is selected. (NOTE: A MARK character cannot be assigned to a menu entry which links to a hierarchical menu.)

You can even set a command key equivalent (Cmd-key box) for each of the menu commands. Try something wild by setting a different colour for each command (only if you have a colour Mac). Your changes will be reflected in the left frame while you are working. Then close your windows and save the changes. That's all there is to it! This applies to all of the Menu List items. They are as follows:

Menu List ID#	Menubar Title
1	Apple (DAs)
2	File
3	Edit
4	View
5	Special
12	*File
15	*Special
16	*Colour

* These items are used by colour Macs and the Mac Portable only.

Desktop icons...

Have you been plagued by one of the following scenarios with respect to the icons on the Desktop?

■ You just downloaded a *StuffIt* file (.sit) from your favourite BBS and decompressed the file. All you see on your Desktop is a generic Mac icon,

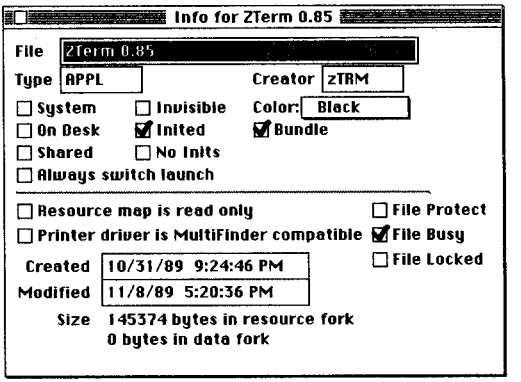
or...

■ You just rebuilt the Desktop and some of the great icons you had previously have now been replaced by generic Mac Icons...

"What we have here is a failure to communicate..." Pardon the cliché, but in essence, that's what is really happening... a failure of the application, cdev, init, etc., to communicate its ICN# resource to the Desktop File.

THE FINDER USES THE DESKTOP FILE TO DISPLAY THE ITEM'S PARTICULAR ICON!

The Desktop File stores information about every application, init, cdev, file, and the like in a resource list, similar to the resource list you find with any file, init, application, item, etc. The Desktop gets this information from each item, by the Finder reading information about the file and 'seeing' what bits are set (type, creator, and others), and then updating the Desktop File.



Depending upon the information stored here, the Finder then goes deeper into the file and looks at the various resources it contains. In essence, if the Finder sees the Bundle (BNDL) Bit set, it then goes in and reads the BNDL, FREF, ICN#, and signature (AKA 'private') resource (and other resources we won't discuss at this time) and checks to see if the Desktop File already has this information. If it doesn't, it adds this info to the Desktop File's resource list. If it already has this info, it just bypasses it. If the Bundle (BNDL) Bit is not set, the Finder will not look for the related resources.

Let's say that you have an older version of a program, and it has a great icon. Now you download a newer version and delete the old version. No problem yet; the new program displays the icon, the same way the old program did. But, unknown to you, the author of the new version forgot to set the Bundle (BNDL) Bit. At this time, however, the Desktop File still has the information from the old version, and since the type and creator are still the same, the Finder doesn't need to look any further. Now comes the problem. You rebuild the Desktop File (holding down the [command] and [option] keys on startup – see below for the reasons why). Rebuilding the Desktop wipes out all information and the resource list it contained, and the Finder scans your directory, looking at every item and builds a new Desktop File. Remember, if the Bundle (BNDL) Bit is not set, the Finder will bypass the related resources and assign a generic icon to it.

Now that you understand (I assume) what's going on, the solution is easy! Via *ResEdit*, open the particular item and check for the BNDL, ICN# and FREF resources. If they are there, close the item, and while it is still selected, do a Get Resource Info from the Resource menu. Then just click on the Bundle (BNDL) Bit to set it. Close all open folders saving changes and quit *ResEdit*. If the folder containing the item was open on your Desktop, just close the folder and reopen it. There may be a short delay, due to the Finder adding this new information to the Desktop File's resource list. You should now see the item with its proper icon.

The same holds true for new items and *StuffIt*. After I download a .sit file and decompress it, if there is a generic icon, I first select the item and close the folder containing it. I then reopen the folder. (Note: if you decompress a .sit file in an open folder, chances are, the Finder hasn't recognised the new file and made its updates to the Desktop File. Closing and reopening the folder alerts the Finder to update the

Desktop File.) If the icon is still generic, I launch *ResEdit*, open the file and check its resource list. I do a Get Resource Info. I then set the Bundle (BNDL) Bit (if the needed resources exist in the file's resource list).

REBUILDING THE DESKTOP FILE WHILE SAVING THE GET INFO COMMENTS...

"Why should I rebuild the Desktop?" ([command] [option] keys)

As time goes on, the Desktop File becomes too large. You will notice this when you create new documents or change things around. What used to take a second or two now may take as long as 20 to 40 seconds. The Desktop File stores information about every item in your directory and retains this information permanently. Even if you delete an old version of a program (or all versions), the information about that item remains intact in the resource list of the Desktop File. Eventually, the Desktop File will become so large, it will slow you down to a snail's pace. When you rebuild the Desktop File, all the information and its list of resources are trashed. Thus, the Finder builds a new Desktop File, only now it's much smaller since many of your old files no longer exist.

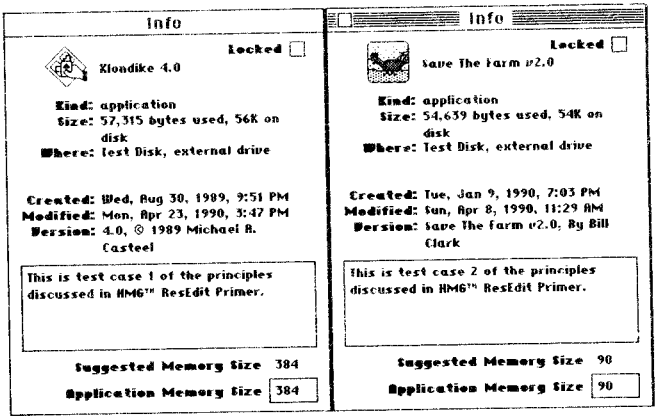
When you rebuild the Desktop File, any information (comments) you added to the item (via Get Info) will be lost! There are some utilities out there (via BBSs) that will rebuild the Desktop File while still retaining your comments. However, I have found a relatively simple way to restore the comments in the Get Info Box after rebuilding the Desktop File, via *ResEdit*!

You will not be able to open the Desktop File via *ResEdit* if you are running under MultiFinder. Set your StartUp to Finder and reboot. Then you can open the Desktop File via *ResEdit*.

The Desktop File will never appear on your desktop. It's invisible. It can only be seen with *ResEdit*, or some other file editing utility which has the capability of viewing invisible files. While in *ResEdit*, if you do a Get Info for Desktop on it, you will see the reason why. The invisible bit is set. UNDER NO CIRCUMSTANCES WHATSOEVER SHOULD YOU CHANGE THIS! It's invisible so that it won't accidentally be deleted. If you delete it, the Finder will construct a new Desktop File. However, you're in for a world of trouble if you had previous intentions of saving your Get Info comments!

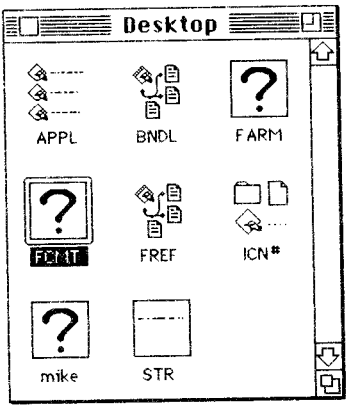
In order to save the comments in the Get Info dialog box, you must first understand where these comments are stored. Why of course, they are stored in the Desktop File in their own special resource called, FCMT, which stands for... Finder Comments. When the

Desktop File is rebuilt, the resource FCMT gets deleted (don't complain to me, complain to Apple). If we save this resource to another file first, we can then paste it back into the Desktop File after it is rebuilt. To prove this theory, I've used a floppy disk to work on (every disk has a Desktop File, even if it's empty). I copied two applications to one disk and added comments to their Get Info boxes:

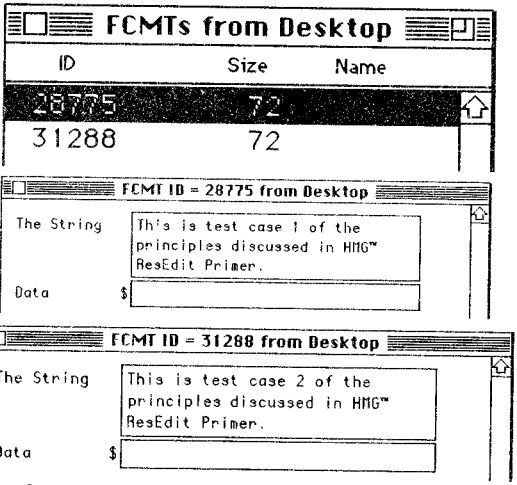

Info
Locked
Klondike 4.0
Kind: application
Size: 57,315 bytes used, 56K on disk
Where: Test Disk, external drive
Created: Wed, Aug 30, 1989, 9:51 PM
Modified: Mon, Apr 23, 1990, 3:47 PM
Version: 4.0, © 1989 Michael A. Casteel
This is test case 1 of the principles discussed in HM6™ ResEdit Primer.
Suggested Memory Size 384
Application Memory Size 384
Info
Locked
Save The Farm v2.0
Kind: application
Size: 54,639 bytes used, 54K on disk
Where: Test Disk, external drive
Created: Tue, Jan 9, 1990, 7:03 PM
Modified: Sun, Apr 8, 1990, 11:29 AM
Version: Save The Farm v2.0, By Bill Clark
This is test case 2 of the principles discussed in HM6™ ResEdit Primer.
Suggested Memory Size 90
Application Memory Size 90

Comments added to Get Info...

I then launched *ResEdit* and opened the Desktop File...

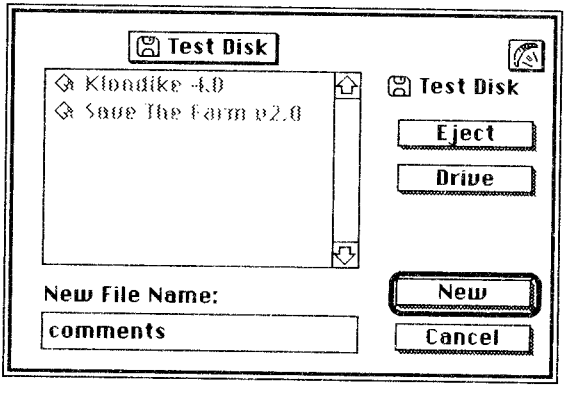


Just out of curiosity, let's look at the FCMT Resources...

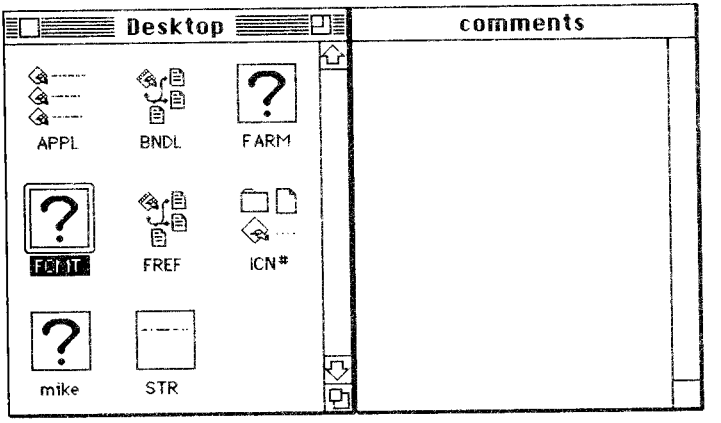


Notice that these contain the comments we added to the Get Info boxes on the Desktop before launching *ResEdit*. Every comment you add to every file will cause the Finder to create a new FCMT Resource ID in the Desktop File. When you change a comment, the Finder merely updates the Desktop File. Now, in order to save this information, what we need to do is create a new file and copy the FCMT resources to it...

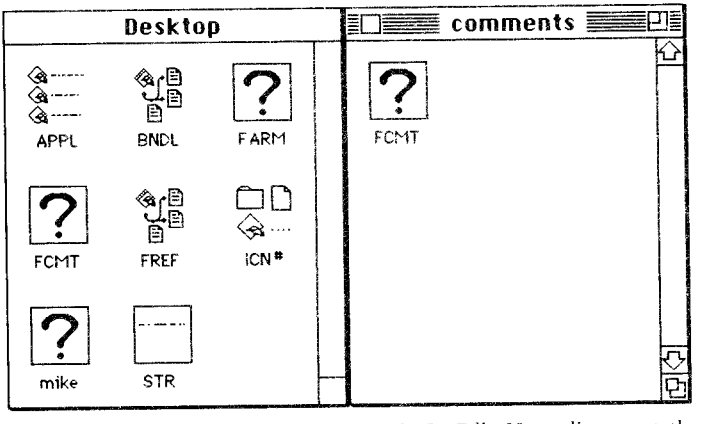
From the File menu select New...



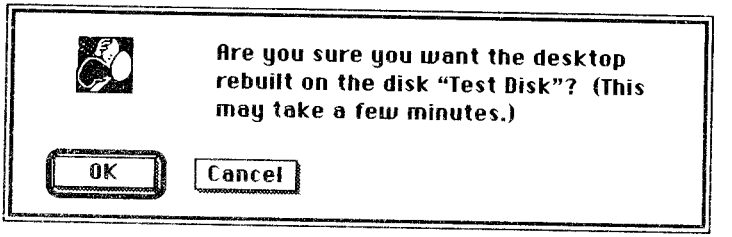
Name the file 'comments file' and click New...



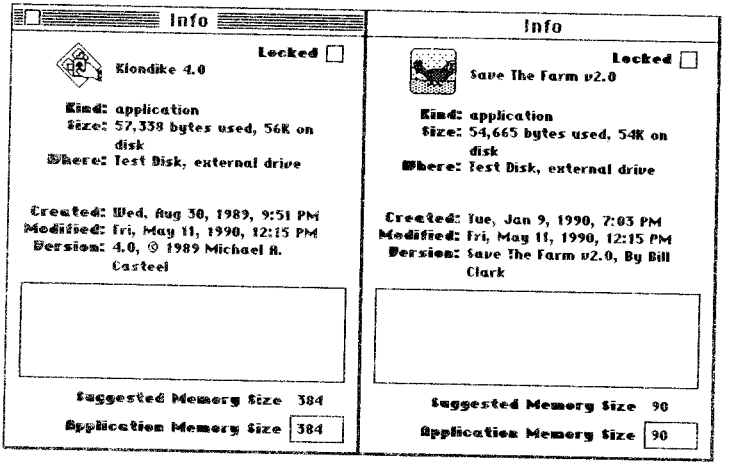
Select the FCMT resource from the Desktop File and then Copy it (Edit menu). Next, Paste it into the comments file...



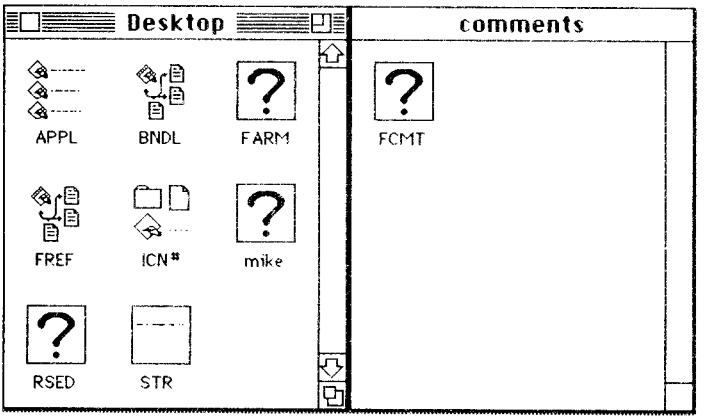
Close all windows saving changes. Quit *ResEdit*. Now dismount the floppy disk by dragging it to the trash. While holding down the command – option keys, reinsert the disk. You will be prompted by the alert over the page:



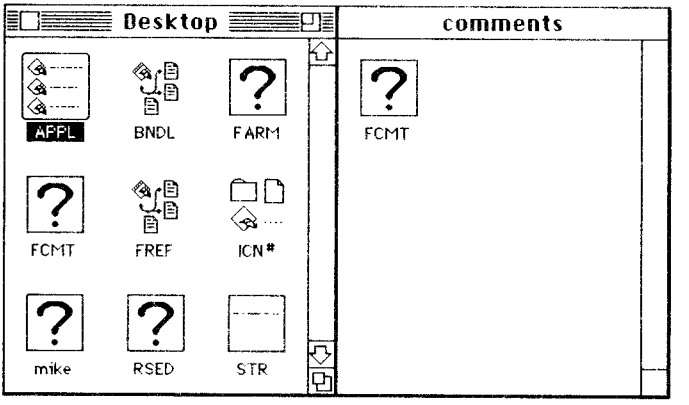
Click on the OK button. Now the Desktop File will be rebuilt. To verify what I said about the FCMT resources being deleted, let's do a Get Info on the file we had previously commented...


Info
Locked
Klondike 4.0
Kind: application
Size: 57,338 bytes used, 56K on disk
Where: Test Disk, external drive
Created: Wed, Aug 30, 1989, 9:51 PM
Modified: Fri, May 11, 1990, 12:15 PM
Version: 4.0, © 1989 Michael H. Casteel
Suggested Memory Size 384
Application Memory Size 384
Info
Locked
Save The Farm v2.0
Kind: application
Size: 54,665 bytes used, 54K on disk
Where: Test Disk, external drive
Created: Tue, Jan 9, 1990, 7:03 PM
Modified: Fri, May 11, 1990, 12:15 PM
Version: Save The Farm v2.0, By Bill Clark
Suggested Memory Size 90
Application Memory Size 90

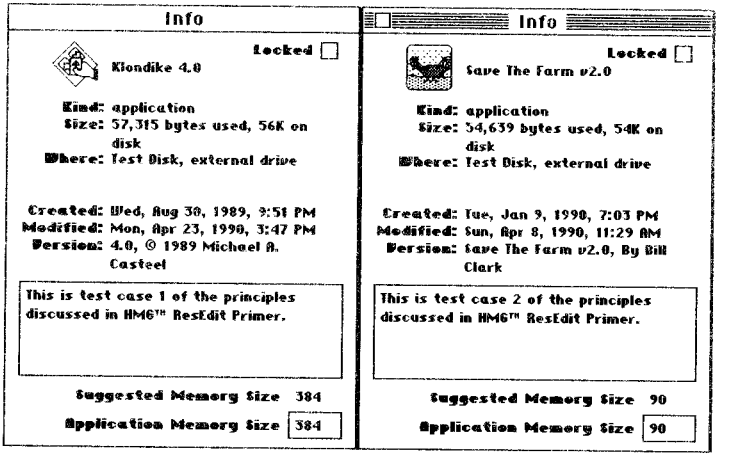
Notice that there are no longer any comments. Now all we have to do is restore the FCMT resources back into the Desktop File. Launch *ResEdit* and open both the Desktop File and comments file on your rebuilt disk...



Now copy the FCMT resource from the comments file...



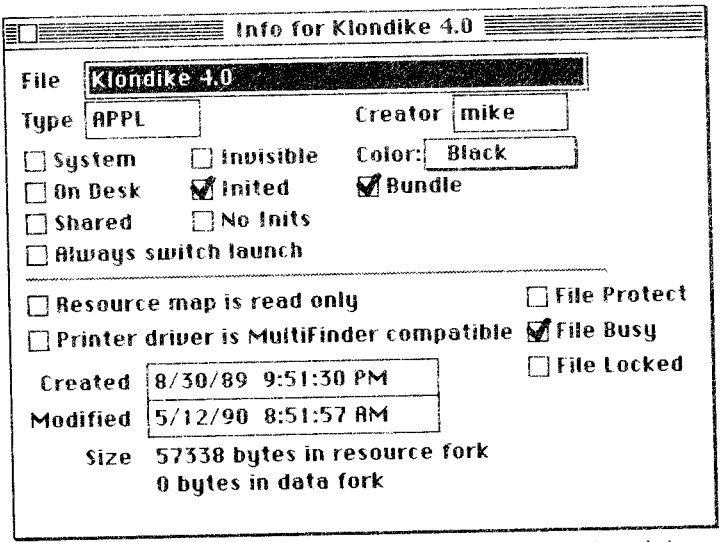
and paste it back into the rebuilt volume's Desktop File. Save all your changes, and quit *ResEdit*. Now do a Get Info on any file that originally had comments, and they should be restored...


Info
Locked
Klondike 4.0
Kind: application
Size: 57,315 bytes used, 56K on disk
Where: Test Disk, external drive
Created: Wed, Aug 30, 1989, 9:51 PM
Modified: Mon, Apr 23, 1990, 3:47 PM
Version: 4.0, © 1989 Michael A. Casteel
This is test case 1 of the principles discussed in HM6™ ResEdit Primer.
Suggested Memory Size 384
Application Memory Size 384
Info
Locked
Save The Farm v2.0
Kind: application
Size: 54,639 bytes used, 54K on disk
Where: Test Disk, external drive
Created: Tue, Jan 9, 1990, 7:03 PM
Modified: Sun, Apr 8, 1990, 11:29 AM
Version: Save The Farm v2.0, By Bill Clark
This is test case 2 of the principles discussed in HM6™ ResEdit Primer.
Suggested Memory Size 90
Application Memory Size 90

Comments restored...

You can now delete the comments file you had previously created (or save it in case of a future system crash that forces the Desktop File to be rebuilt). That's all there is to it!

About the file info window...



The System bit indicates that the file cannot be renamed, and that a warning will be given when the file is dragged to the trash.

If Invisible bit is set, the file is not displayed by the Finder. *ResEdit* can see it, however, as well as Finder substitutes like DiskTop and Disk Tools.

On Desk flag means that the file is on the Desktop (not in any folder).

Inited bit is set by the Finder when it determines the file's location and window.

Bundle flag is set by the Finder if you have a BNDL resource in the file.

If the shared flag is set means the application can be opened more than once.

No Inits concerns the control panel's search for INITs in cdev files. If this bit is set, the control panel will skip the file in question. This

speeds the system startup as it does not need to search all control panel files, only those that actually have INIT resources in them.

Always Switch Launch is used to force the Finder to try to switch startup volumes whenever an application with this bit is set. This is similar to holding down the [option] key while attempting to launch an application. Note that this will not work in MultiFinder or with older Finders.

Resource map is read only if checked, locks the file's resources so that they can't be changed.

Printer driver is MultiFinder compatible is set only for printer drivers that work with MultiFinder.

File Protect flag is not currently used.

File Busy set means the file is open.

File Locked bit locks the File (cannot be renamed or thrown away).

Created is the file's original creation date.

Modified is the date the file was last modified.

Resource fork size is the total size of all the file's resources combined.

Data fork size is the total size of all information stored in the data fork. The data fork may store information about user preferences or anything else the programmer desires.

Editing menus

This chapter provides a summary of the way in which MENU resources are edited by *ResEdit* 2.0. This version of *ResEdit* offers far more control over menus than previous editions, including the use of colour and simple handling of hierarchical menu linkages.

I assume throughout this chapter that you are already familiar with some basic concepts, including the general use of *ResEdit* (and the appropriate safety precautions regarding the use of this potent piece of software), the Apple Colour Picker, and some of the details of the Menu Manager that are discussed in *Inside Macintosh*. For clarity, I'll always use the capital letter word 'MENU' when I'm referring to a menu resource; the lower-case 'menu' will be used for its more generic sense.

One thing to remember throughout this discussion is that a MENU resource is a template which determines only the initial appearance of a menu on the screen. An application can use a variety of toolbox calls to modify the appearance of an existing menu, either in whole or in part. These toolbox calls are detailed in *Inside Macintosh*, in the section on the Menu Manager.

Please note: There is a variety of shareware and commercial products which have special effects on the operation and/or presentation of menus. If you use any of these programs, inits, DAs or cdevs, you should be aware that your menu changes may be overridden by these programs or may cause other problems, up to and including system crashes. Products to watch out for include, but are hardly limited to, the following:

- All macro generators (eg *QuicKeys*, *MacroMaker*), as they affect the processing of keyboard equivalents. This also applies to applications such as Microsoft *Word 4.0* and others which allow you to assign your own keyboard equivalents.
- Programs which enable general setting of menu display colours (eg *Kolor*).
- Programs which perform invisible menu item swapping, such as *Suitcase* and *Font/DA Juggler*.
- Programs which display entries in Font menus in the actual fonts.

EDITING THE MENU DISPLAY

As with most resources, the editing of a menu with *ResEdit* begins with your opening the application or file which contains the resource to be edited. You'll be presented with the usual list of resource types found in the file, from which you will select MENU by double-clicking. You will then be presented with a scrolling list of all MENU resources in the application or file, along with their resource ID numbers and resource names (if any). When you select a specific MENU resource for editing, its contents will be presented in various displays.

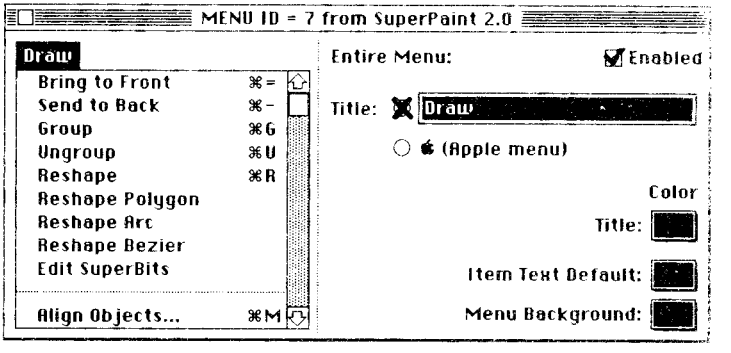
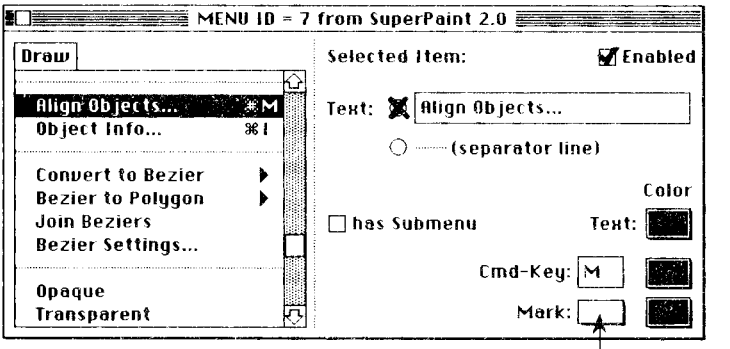
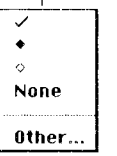


Figure 1 shows a sample of this dialogue box, containing one of the title menus from Silicon Beach Software's *SuperPaint 2.0*. The title shows itself in the menu bar when running the program.





Mark Characters

Figure 2 – Menu editing dialogue box (Item)

Figure 2 shows a sample of the dialogue box when an item is selected. Notice how the appearance of the dialogue box has changed (refer to Figure 1).

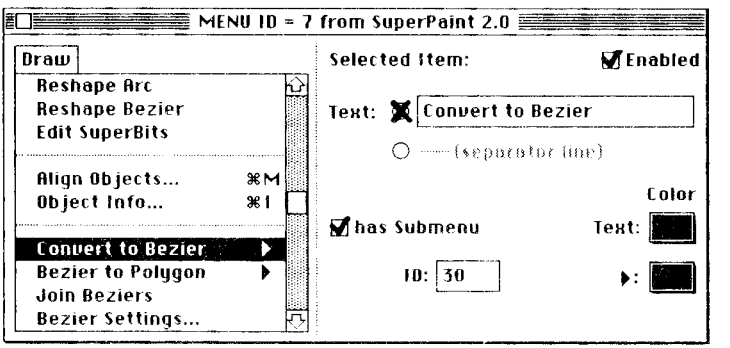


Figure 3 shows an example of how the window will look when an item has a submenu linked to it. Again, the menu dialogue box has changed in appearance.

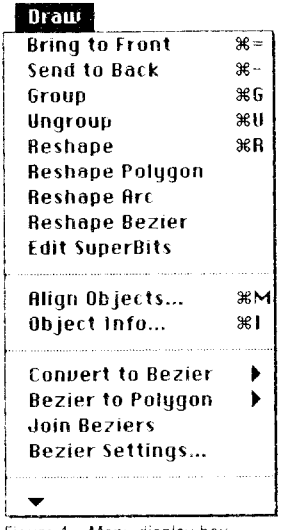


Figure 4 – Menu display box

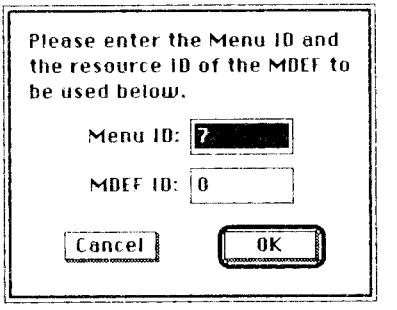


Figure 5 – Setting ID#s

Figure 4 shows the sample display for the menu being edited in the first screen. Note that not all of the menu items are visible in the sample, which is limited to eighteen items. The scroll-down arrow is operable within the display menu and will display the remaining items. While it is possible to click on items in this menu, this will have no effect. However, all other changes you make regarding colours, marks, icons and other aspects of the individual menu items will be shown within this display. Figure 5 shows the dialogue box in which to set or change the MENU resource ID# and the MDEF ID#. These will be discussed later. Figure 6 shows the submenu selected above.

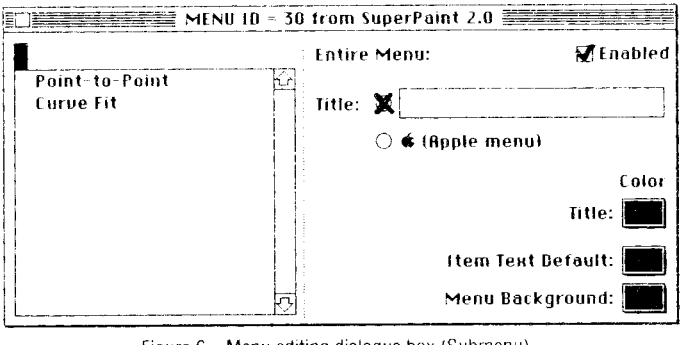


Figure 6 – Menu editing dialogue box (Submenu)

EDITING THE OVERALL MENU PARAMETERS

• Title (Figure 1):
The title box is a text entry field which holds the text, if any, that will be placed on the menu bar in order to identify this menu when it is displayed. This is an optional field; if the MENU being edited will be a hierarchical menu, there should be no title.

• Menu ID# (Figure 5):
This is the resource ID number assigned to the MENU resource being edited. Note that if the MENU is also to have a resource name, you cannot add it here. To give the MENU a name, use the Get Info function in *ResEdit*.

• MDEF ID (Figure 5):
This field holds the ID of the Menu DEFinition routine to be used by the Menu and Resource Managers whenever this MENU resource is being presented, modified or otherwise accessed by an application. This code handles the actual sizing and display of the menu, as well as such special features as scrolling and popup menus. In almost all circumstances, you will leave this parameter set to zero, thus using the default MDEF supplied with the system. If you want to create your own MDEF, you'll need to do so in PASCAL, C, or Assembler, and will need to be familiar with the operation of the Menu Manager. *Inside Macintosh* has more details. However, if you have created your own MDEF, the simplest way to test it is to copy it into an application with existing MENUs and substitute its ID# in this field.

• Enabled (Figures 1,2,3,6):
This check-box field determines whether the selected item will be enabled when the menu is displayed. If checked, the menu item is enabled. If blank, the item is disabled and will be displayed in its greyed-out form.

• Colouring the Overall Menu (Figure 1):
The three boxes on the lower right of the editing window (under Colour) show the colours to be used when the menu is initially displayed. When you select an item from the menu list, these three boxes change (Figures 2,3,6) and will reflect how the colouring will apply to the items directly adjacent to the colour boxes. If you click on one of these boxes, you will be presented with the Apple Colour Picker to select a new colour. If you're on a monochrome system, you will not be able to use the Colour Picker to assign colours.

GENERAL OPERATIONS ON MENU LIST ENTRIES

Selecting an item to be edited: by clicking on an entry in the scrolling list on the left side of the editing window, you will select that menu item for editing. It will be highlighted, and will remain highlighted until you click on a different entry in the list, no matter what else you do in the editing window. Note that *ResEdit*'s Cut, Copy, Paste and Clear commands will operate on selected menu items as long as the text-entry cursor (I-beam) is not visible on the screen; otherwise, *ResEdit* assumes you're editing text.

Adding new menu items: if you want to add an item to a menu, simply choose Create New Item ([command] K) from *ResEdit*'s Resource menu. A highlighted blank line will appear at the end of the scrolling list of items. Place your cursor in the text entry box and click. You will now have an insertion point to enter your text for the new item. Note that if you wish a line drawn across a menu, check the separator-line radio button. **Warning**: be extremely careful when checking the separator-line radio button. If you have an item selected with its text in the text editing box, and you check the separator-line radio button, the item and its text will be replaced by a separator line.

Re-ordering menu items: to re-order items in the list, use *ResEdit*'s Cut, Copy, Paste and Clear commands while the text-entry cursor (I-beam) is not visible on the screen.

Deleting a menu item: select the menu item and choose Clear Item from the Edit menu.

EDITING MENU LIST ENTRIES

When a menu list entry has been selected as described above, you can edit its appearance with the various check boxes, radio buttons and text fields provided on the lower-right side of the editing window. This section provides details on these features of the *ResEdit* MENU editor. Because of their complexity, hierarchical menu linkages and icons are treated in separate sections.

Note that there is no method for editing the font or font size used to display menu item text. These items are displayed in the system font, normally 12-point Chicago. Special toolbox calls are available to change the system font on a temporary or permanent basis. You may

use these calls within an application to modify a menu-text item's appearance.

Style: the Style menu will allow you to assign a type style for the item being edited. Available styles include Bold, Italic, Underline, Outline, Shadow, Condensed and Extended. As is usual on the Mac, you may combine these styles as you see fit.

Mark (Figure 2): this pull-down menu box, along with its adjacent Colour button, is used to define a mark character – a single character which is used to flag a menu item which has been selected by the user. For example, most word processors which have a menu listing the available fonts, use the check-mark ([control] R) character to indicate which font a user has chosen. After a menu item with a mark character assigned has been selected, the mark character will appear in the menu to the left of the menu item's text. The next time that item is selected, the mark character will be removed from the menu display. The Colour button to the right of the Mark boxes is used to specify the colour of the mark character; it defaults to black. Note: a mark character cannot be assigned to a menu entry which links to a hierarchical menu.

Cmd-Key: this text-entry box, along with its adjacent Colour button, is used to specify a keyboard equivalent for the menu item; ie a key-stroke which will have the same effect as selecting the menu item from the menu. This keystroke always involves the [command] key plus the 'selected' keyboard character. Shifted and unshifted characters are treated as equivalent; thus 'c' and 'C' are treated the same, as are '=' and '+'.

Notice: please adhere to the following Apple guidelines:

1. Apple recommends that letters always be specified as upper-case for keyboard equivalent purposes. There is one critical exception to this rule. If you specify a number for a keyboard equivalent, it will only operate in the unshifted state. [Command]-[shift]-[number] combinations can never be keyboard equivalents, as they are instead used to trigger FKEYs.

2. Apple requires you to use the standard keyboard equivalents Z, X, C and V for the editing commands Undo, Cut, Copy and Paste. If you modify these, editing may fail while using desk accessories.

To assign a keyboard equivalent to a menu item, enter the actual character in the Cmd-Key text box. Once you make an entry, its match will appear in the menu list. Keyboard equivalents are displayed on the same line as the menu item's text, to the right of the text, preceded by the command character. The Colour button adjacent to the box may be used to set the colour in which the keyboard equivalent is displayed. Note: a keyboard equivalent cannot be assigned to a menu entry that has an associated icon or which links to a hierarchical menu.

CREATING MENU RESOURCES FOR POPUP MENUS

A relatively recent feature available to developers is the popup menu. This feature allows a menu to appear anywhere on the screen, without any attachment to the menu bar. Popup menus are used primarily to present the user with a set of choices. When a popup menu is displayed, only the currently-selected choice is visible. When the user places the cursor over this choice and presses the mouse button, the remainder of the menu appears and the user may make a new selection. In terms of MENU resources, the only distinction between the MENU for a popup and the MENU for a menu-bar menu is that the Title field of the popup menu's MENU is left blank.

CREATING MENU RESOURCES FOR HIERARCHICAL MENUS

Most menu items are used for one of three purposes: to select an operation to be performed (eg Paste, Quit, Check Spelling); to indicate a selection from a group of choices, such as a font or type size; or to enter a process in which the user will specify parameters or make multiple selections (eg Page Setup..., Print...). In the last case, the application usually presents the user with a dialogue box. Typically, this dialogue box includes one or more text fields, selection lists, and/or sets of radio buttons with which the user can provide necessary information so that the program can proceed to perform some action. Usually, such menu choices are followed by either three periods or the ellipsis character ([option];), which combines these three periods (...) into a single character.

In some cases, however, the user merely needs to make a single choice from among a small set of alternatives. In this case, bringing

up a dialogue box is a slow and wasteful process (and so is the time spent in designing it!). Instead, a hierarchical menu is used. When the user highlights a hierarchical menu item on a menu, a second menu appears on the screen. This new (second-level) menu is normally placed to the right of the first menu, but will be placed to the left if there is insufficient space. A menu item which links to a hierarchical menu is normally indicated by a right-facing arrow placed in the position normally occupied by the keyboard equivalent. The first menu item of the second-level menu is placed adjacent to this arrow.

If the user moves the cursor vertically on the original menu, the second-level menu disappears. However, by moving the cursor onto the second-level menu, the user is now able to select an item from this menu. Items on a second-level menu may include links to lower-level hierarchical menus, and so on down the line.

The actual creation of a hierarchical menu linkage is quite simple. Select (or create) the menu list item which will link to the lower-level menu. Click on the Has Submenu box. From the Resource menu, select Open Submenu. From the Resource menu select Create New Item. Create all the items you want in the submenu, and set their attributes. Now close the submenu dialogue box. The two are now linked! Note: A hierarchical menu entry in a menu list can have neither a mark character nor a keyboard equivalent associated with it. Also, remember that early versions of the Macintosh System Software do not support this feature; however, you should be safe with System 4.1 or later.

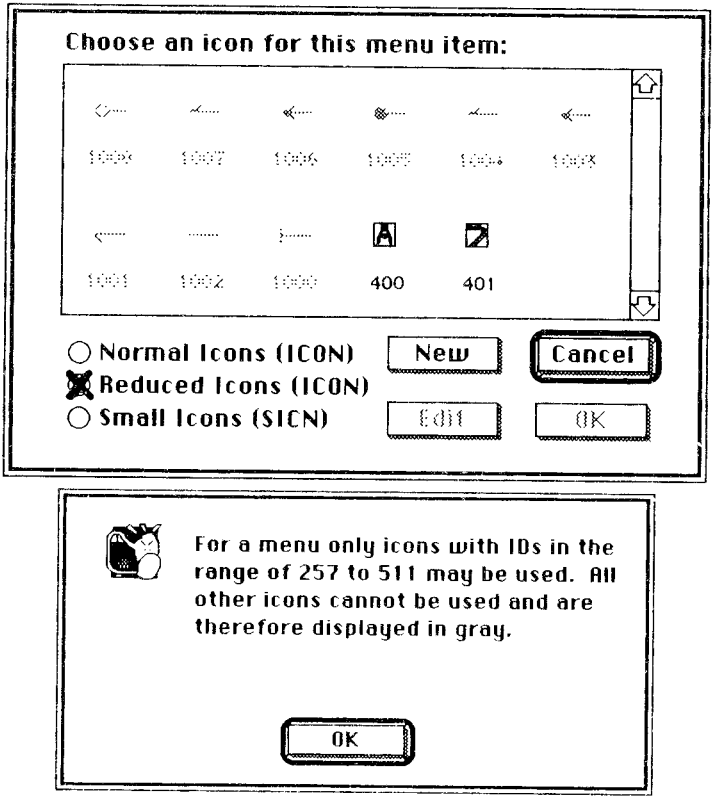
ADDING ICONS TO A MENU

In addition to all the other features described above, it is possible to attach regular- and reduced-sized icons, as well as 'small icons' (SICNs), to an individual menu item. The icon is displayed to the immediate left of the menu item's text, with the text centred vertically relative to the icon. If a mark character is associated with the menu item, and the item has been selected, the mark character will appear to the left of the icon. Note that when the menu is displayed, a regular-size icon will effectively cause this menu item to require the space normally needed by three 'standard' menu items. Notice also that if you use a reduced-size ICON or an SICN, you will not be able to assign a command equivalent to the menu item.

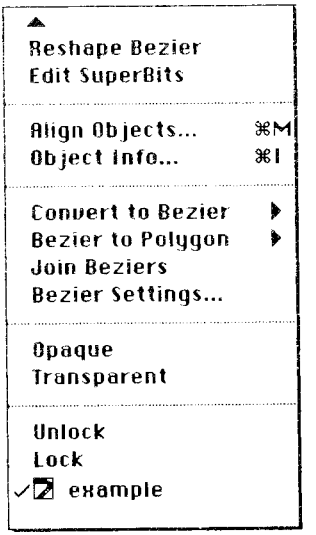
To attach an icon or small icon to a menu item, you must first make sure that the appropriate resource is available. For regular- and reduced-sized icons, you must copy the appropriate ICON resource into the current resource file. For a small icon, you must copy the appropriate SICN resource into the current resource file. For either type of resource, you must also assign a unique Resource ID# that is between 257 and 511.

Once this is done, select the MENU resource to be edited, and click on the desired menu item from the selection list. Then follow these simple steps:

1. From the Menu menu, select Choose Icon...


Choose an icon for this menu item:
400
401
Normal Icons (ICON)
Reduced Icons (ICON)
Small Icons (SICN)
New
Cancel
Edit
OK
For a menu only icons with IDs in the range of 257 to 511 may be used. All other icons cannot be used and are therefore displayed in gray.
OK

2. Select your icon (or create a new one) from the Icon Display box and click the OK box...



Your icon should now appear in the Menu Display box.

TIPS ON COLOURING YOUR MENUS

If you have the Kolor cdev installed, you should note that the colours set by this init will not appear in the boxes within the editing window. They will also not be used in the sample menu display. However, any colour which you assign as a default value will have that default overridden by the colours you specify with Kolor at the time the menu is actually used.

When you set colours for menu elements, be sure to pay attention to the degree of contrast between item colours and the background colour. This contrast should be checked in both enabled and disabled states, as the latter is generally harder to see. It's also a good idea to be certain that the selected colours provide sufficient contrast in all possible display modalities (ie black-and-white, greyscale, and 2-, 4-, 8-, 24- and 32-bit colour modes). One very striking approach is to use a 'reversed' colour scheme. Set your background colour to a very

dark shade, and use light colours for item, mark and command colours.

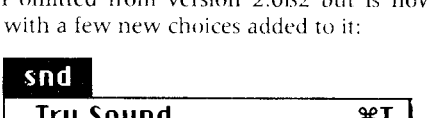
While it is possible for every element of a menu to have a different colour, remember that too many colours can be extremely distracting to the user. In general, colour is best applied within menus as a means of highlighting critical information, rather than a mechanism to show off what the monitor can do. If you want to make colourful art, get a good paint or draw program and have fun that way!

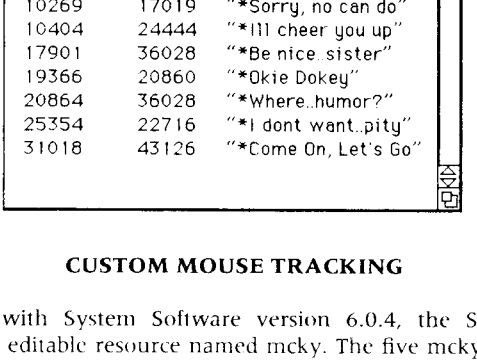
Bits & pieces...

This chapter is similar to a 'junk drawer' in a desk; where one would keep small items of value. New techniques, features and uses for *ResEdit* that I have come upon are placed here.

SOUND (snd) RESOURCES

ResEdit 1.2 thru 1.3d1 had the ability to play sounds (snds). This feature had been omitted from version 2.0ß2 but is now restored in version 2.1ß5 with a few new choices added to it:




snds from Simpson.snds
ID Size Name
8759 11260 "*EatMyShorts"
10269 17019 "*Sorry, no can do"
10404 24444 "*I'll cheer you up"
17901 36028 "*Be nice..sister"
19366 20860 "*Okie Dokey"
20864 36028 "*Where..humor?"
25354 22716 "*I dont want..pity"
31018 43126 "*Come On, Let's Go"

CUSTOM MOUSE TRACKING

Beginning with System Software version 6.0.4, the System file includes an editable resource named mcky. The five mcky resources numbered 0 to 4 correspond to the five speed choices, Tablet to Fastest, in the Mouse Control Panel. By using *ResEdit* version 2.0ß2 or later, you can edit the resources and adjust the mouse's responsiveness to suit your work habits.

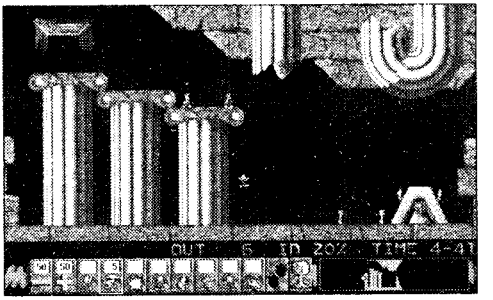
Each mcky resource has eight threshold levels that determine how mouse movements affect pointer motion on the screen. For instance, the fastest speed, resource mcky 4, is preset to thresholds of 1, 4, 7, 10, 13, 15, 16, and 255. I reset those values to 1, 2, 4, 4, 8, 9, 10, 11. Acceptable values range from 1 to 255, with lower numbers generat-

Subscribe to MACFORMAT!

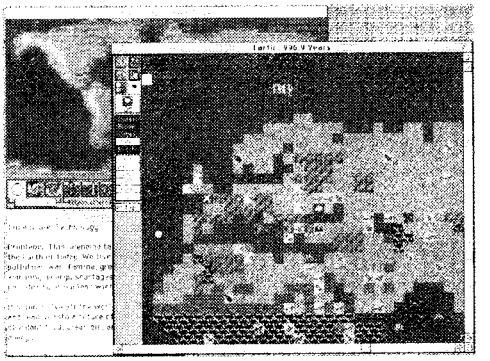
If you take out a 12 month subscription to MACFORMAT, you not only get Britain's leading Mac monthly through your door every month, you get special subscribers' mail order deals and a free gift worth over £35.

You can choose from *Adobe Type Manager*, *Lemmings* or *SimEarth*, and you get 12 issues packed with more than a thousand pages of hints and tips, plus cover disks with over 2Mb of software every issue, all for just £47 (UK)!

Phone the subscriptions hotline number, 0458 273279, quoting ref MFBOOK to get your subscription started!



Lemmings



SimEarth

ResEdit is extremely powerful software used by programmers to create Macintosh applications. But it can also be used subtly to alter those programs: changing icons, editing menu items, adding keyboard shortcuts, copying sounds...

In this book we introduce some of the concepts of *ResEdit*, so you can customise your Mac and its applications just the way you like them!

WARNING: Because *ResEdit* is such a powerful program, it can easily damage programs and System files. Always work on a duplicate copy, and have a backup.

A MACFORMAT BOOK

from

Given free with **MACFORMAT** April 1994. Not to be sold separately.

MACFORMAT